AD-A039 179 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 17/2
AN ECL GATE ARRAY REALIZATION OF A COMPUTER FOR REAL TIME SPEEC--ETC(U)
MAR 77 R BERGER F19628-76-C-0002
UNCLASSIFIED TN-1977-11 ESD-TR-77-58 NL

| OF |
ADA039179

END
DATE
FILMED
5 - 77

ESD-TR-77-58

Technical Note 1977-11

R. Berger

An ECL Gate Array Realization of a Computer for Real Time Speech Synthesis

7 March 1977

Prepared for the Department of the Air Force
under Electronic Systems Division Contract F19628-76-C-0002 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

AN ECL GATE ARRAY REALIZATION OF A COMPUTER FOR REAL TIME SPEECH SYNTHESIS

R. BERGER
Group 24

TECHNICAL NOTE 1977-11

7 MARCH 1977

LEXINGTON MASSACHUSETTS

(See form 1473)

ABSTRACT

This note describes the Lincoln Integrated Speech Synthesizer (LISSYN), a general-purpose computer intended for speech processing, whose central processor is made from ECL gate arrays (large scale integrated circuits custom built at Lincoln Laboratory).

The goal was to use gate arrays to implement in real time the synthesis portion of a linear predictive vocoder operating at 4800 bits/sec. The design process stressed minimizing the number of different kinds of gate arrays and the number of non-gate-array circuit packages. The result is a general purpose computer structure featuring: single 1024 x 16 memory for data and program, 200 nsec instruction cycle, 950 nsec add/shift multiply, binary serial input, analog output via a 12-bit D/A converter and desampling filter, 0.35 cu. ft. volume, 60 watts DC power, 11 gate arrays of 5 types, 30 memory IC's, 27 other circuit packages. The LISSYN runs the linear predictive speech synthesis in 43% of real time.

CONTENTS

I. INTRODUCTION

A. History

Over the past few years, Lincoln Laboratory has been developing an ECL gate array technology as a means of providing fast turn-around time for the design of custom, high-speed circuits of high levels of complexity. A fixed set of diffusions defines the transistors and resistors of a basic cell array which can be customized by the patterning of two levels of metal interconnect. The array is comprised of an 8x8 matrix of cells each of the complexity of a triple 3-input gate. In cooperation with efforts in digital speech research at the Laboratory, a gate array demonstration project was conceived using the new techniques to fabricate the synthesizer portion of a Linear Predictive Coding (LPC) algorithm[1,2] for digital speech coding. The Lincoln Integrated Speech Synthesizer (LISSYN) was designed and built for this purpose. The LPC algorithm has become prominent due to the simplicity of its implementation in comparison with other narrowband speech coding methods. Since the synthesizer (receiver) portion of the algorithm is simpler than the analyzer (transmitter portion), this device represented a suitable first major application of the gate array technology. Implementation of the entire LPC algorithm would require extra memory and a more complex architecture, requiring more effort and time than necessary to achieve the goals of the gate array development project.

A receive-only processor could be used in speech terminal tandeming experiments and therefore would fill a needed role in current speech research while demonstrating the gate array capabilities and revealing needed improvements in processing techniques. In addition, potential systems applications for stand-alone LPC synthesizers have been identified. One means of providing a free-form type of voice conferencing capability is to sum the analog speech signals of all conference participants at a central point, and to distribute the result back to the individual speakers. When vocoders are used, this method requires that a number of synthesizers be provided at the inputs to the central summing junction, followed by a single vocoder analyzer at its output. In this way all voice traffic to and from the analog summing point is digital. Although the voice quality of a system of this type is currently not as acceptable as that of some other forms of digital conferencing, one reason it has not been too seriously considered is the assumed high cost of multiple LPC synthesizers. The demonstration of a potentially inexpensive implementation of such devices using LSI technology is therefore valuable in that context. A second application for a stand-alone LPC synthesizer is in the case of wideband-narrowband interoperability. A currently favored solution to this problem includes the use of vocoder tandems, in which a 2.4 Kb/s LPC stream is converted to analog form and redigitized at a 16 Kb/s rate by a CVSD encoder. This particular tandem suffers from severe quality degradation, while the reverse tandem (16 Kb/s to 2.4 Kb/s) yields more acceptable results. One could dispense with the LPC to CVSD conversion if wideband users could accommodate 2.4 Kb/s LPC data directly. This requires the use of an LPC synthesizer in the wideband facility, thereby increasing the cost and size of the terminal equipment. Again, the demonstration of a potentially small and inexpensive LSI approach to this problem is an appropriate exercise.

B. Basic Architecture

The high speed of ECL circuits permits the retention of architectural simplicity rather than the use of tricks such as pipelining to achieve adequate speed. In order to simplify design and construction of the LISSYN and minimize the number of gate arrays required, while including almost all logic except that intrinsically unsuitable for ECL gate arrays (mostly memory and TTL), an add/shift multiplier using a hard-wired control signal sequence was adopted instead of the use of separate multiplier hardware. Although the multiply operations occupy 42% of the LPC synthesizer processing time and no other processing could be done in parallel, the overall speed proved to be more than twice that required for real-time speech synthesis.

The second architectural decision hinged on the availability of a 1024x1 ECL RAM. Even though there turned out to be only 85 words of dynamic storage needed for the LPC synthesis algorithm, it cost little more in money and power and no more in space to use 1024x1 ECL RAMs instead of 256x1 ECL RAMs. The 1024-word memory was then large enough for instructions and tables of constants, as well as for dynamic storage. Since memory is the principal use of extra packages beyond gate arrays, it was decided to build the LISSYN with a single memory, a 1024x16 ECL RAM.

Once this decision was made, three other architectural features followed:

1. There would be no overlapping of instruction fetch and operand fetch/store to speed up the machine.
2. Instruction words would be only 16 bits long, and therefore a control memory would be needed to decode the instructions.
3. Since the LISSYN was required to operate in a stand-alone mode as an LPC synthesizer, a separate non-volatile memory was needed

to store an image of the LPC program and tables. Since this memory could be slow, and since at that time TTL ROM's were available with 16 times the bit density of ECL ROM's, it was decided to implement that memory with four 1024 x 4 TTL ROM packages. Even though it cost eight other packages to interface the image ROM with the LISSYN, the overall package count was less than that for using ECL ROM for program and tables and the resultant machine was more versatile. Testing was also made much easier, since the program portion of memory didn't have to be replaced by a dynamic image memory in order to run diagnostics. It would not have been trivial to plug in such an image memory. Cable delays alone would have compromised the LISSYN timing.

The resultant LISSYN architecture is shown in Figure 1 as a block diagram organized to show the balance between gate arrays and other packages. The number of packages needed for each block is shown circled. Not included is the tester, a detachable box which is used for hardware and software debugging. The central processor is made almost entirely of 11 gate arrays of 5 different types: four 4-bit ALU slices, four 4-bit Register Transfer slices (containing and connecting the remaining general registers), two control gate arrays, one timing phase generator. Only 8 commercial 16-pin ECL DIP's were needed to complete the central processor logic, including interfacing with the tester, with the ECL memories, and with the panel switches.

The main memory has 1024 16-bit words of ECL RAM and 32 16-bit words of ECL ROM. The latter is used by the tester and also holds a bootstrap program that loads the image memory into the RAM. The three memories, main, control, and image, use 30 IC packages. Translations between ECL and TTL take 9 packages, other TTL logic 7 packages. The remaining three packages are special devices: a 20 MHz clock, a 12-bit D/A converter, and a modem inter-

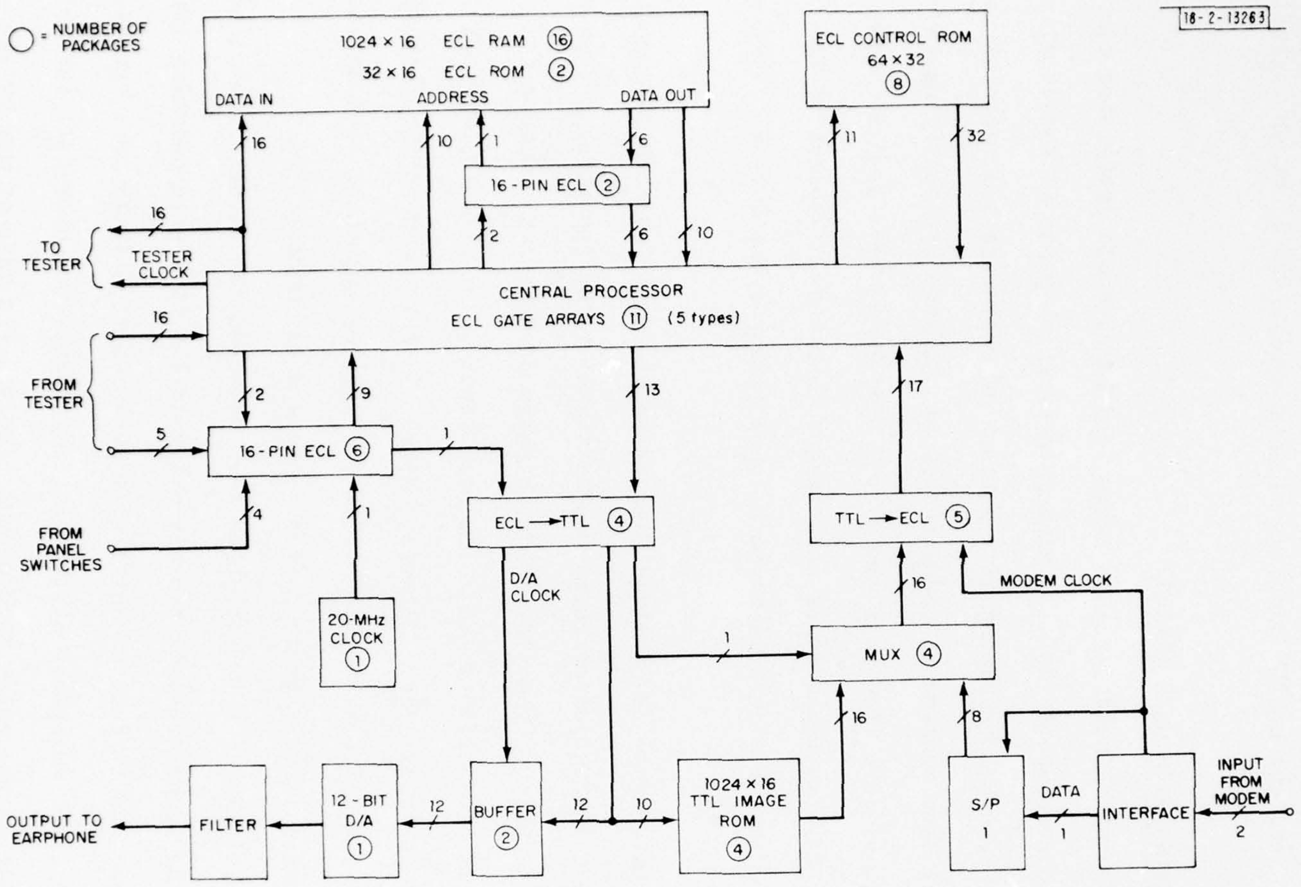


Fig. 1. LISSYN organized by package type.

face. Altogether, the LISSYN contains 11 gate arrays and 57 other logical packages.

II. DESIGNING WITH GATE ARRAYS

The Lincoln Laboratory gate array is a large-scale integrated circuit employing emitter coupled logic[3]. Each gate array chip has 64 essentially identical cells, each consisting of a pattern of resistors and transistors. The gate array user configures the chip for his purpose by specifying the metal connections that transform these cell components into logical elements and then connects the logical elements into larger functions. Typical of the logical complexity of a cell is three 3-input gates or one D-type master-slave flip-flop.

Each gate array is provided with 24 input amplifiers and 24 output drivers. Input and output circuits can be chosen to be inverting or non-inverting, and their voltage levels match those of MECL 10K. Output signal pins can be sacrificed to allow more inputs, but input amplifiers are not available for these extra signals. Since the threshold voltage interior to the gate array differs slightly from the MECL 10K threshold used on the input amplifiers, extra input signals suffer reduced noise margin.

Gate delays of 0.65 nsec have been measured for lightly loaded gates within an array[3]. However, an average delay of 1.5 nsec/gate, counting input and output drivers, was typical of multi-gate paths on the LISSYN gate arrays.

The salient advantage of the gate array approach over full custom LSI development is the quick turn-around time and the ease of use by the system designer. In the ideal case, the wafers already exist with all diffusion steps completed. The system designer chooses logical cell configurations (e.g., a master-slave flip-flop) from a cell library and connects them on a logic drawing into the function he desires, following a few simple loading rules. The total elapsed time for the fabrication process is eight weeks[3].

In practice, it was necessary for the LISSYN system designer to become more closely involved in the details of the gate array production. Some examples of this involvement were:

1) A test program must be developed to automatically test completed gate-array wafers. Close cooperation between the designer of the array and the developer of the test program was needed.

2) Each gate array is simulated on a wirewrap board with commercial ECL packages before it is produced, partly to check the logical design and partly to check the test program mentioned above. The insight of the designer was found to be useful when it came time to debug these simulators.

3) There were cases where the desired logical cell pattern did not exist in the library. It was then necessary for the designer to work on the transistor level and specify a new logical cell. The major example of a new cell for the LISSYN was a one-cell master-slave flip-flop to replace an earlier two-cell version.

Gate arrays come packaged in 64-pin square ceramic flat packs with 16 leads on .050 inch centers along each edge. The flat packs are then mounted on an ECL wirewrap board (Augat ECL-21-180), designed for 16-pin DIP's, by means of a printed-circuit adaptor board that covers 4.5 of the 16-pin DIP positions.

Cooling of the gate arrays, some of which dissipate 4 watts, requires an air flow of 300 linear feet per minute across a 4-finned cylindrical heat sink which extends 0.5 in. above the ceramic package to which it is epoxy bonded. Figure 2a shows a fully packaged and mounted gate array.

The LISSYN logic (excluding the tester) occupies four of the six sections of the 180-DIP wirewrap board. The fifth section is empty and the sixth holds the audio filter components. The LISSYN dissipates 60 watts of DC power. Figure 2b shows the completed LISSYN wirewrap board.

The full LISSYN system (including fans and power supplies, but excluding the tester) fits into a cabinet 3.5 x 8.5 x 20 inches, a volume of 0.35 cubic feet. The LISSYN cabinet is shown in Figure 2c.

III. INSTRUCTION FORMAT

All LISSYN instructions are 16 bits long and have a simple two-field format:

OP (bits 10-15),	a 6-bit instruction code
Y (bits 0-9),	a numerical field.

The Y field is long enough to address the entire 1024-word RAM, which has octal addresses 0-1777. Most LISSYN instructions which use Y as an address for fetching data from or storing data into memory have both indexed and unindexed versions. In the indexed version, Y is interpreted as a 10-bit positive integer and added to the contents of the index register to form the memory address. In this way it is possible to read the 32-word ROM, which has octal addresses 2000-2037.

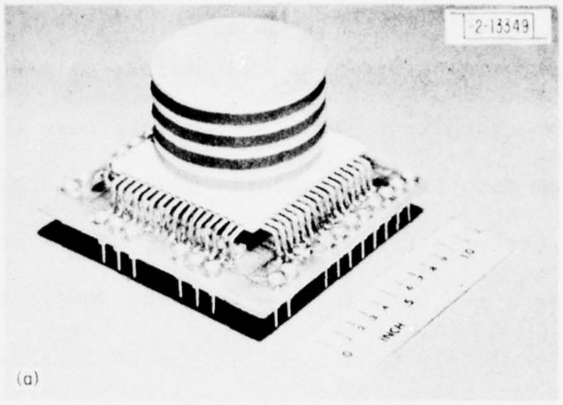


Fig. 2a. An ECL gate array on an adaptor board with heat sink.

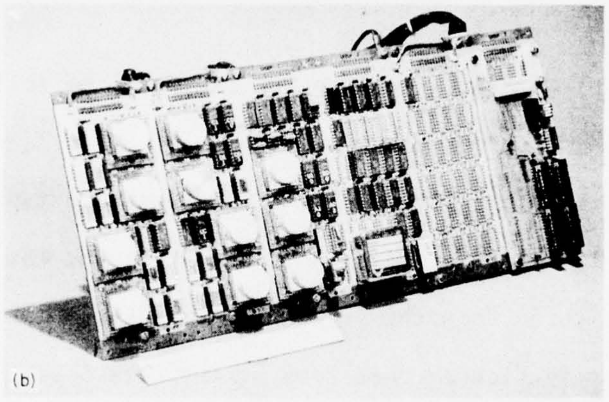


Fig. 2b. The LISSYN wirewrap board.

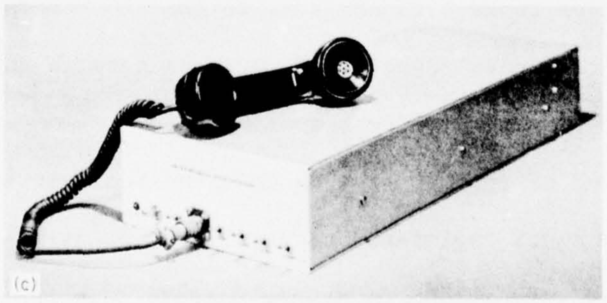


Fig. 2c. The complete LISSYN.

Most jump instructions use Y as the address of the next instruction to be executed if the jump condition is met. In this case Y replaces bits 0-9 of the present program counter, leaving bits 10-15 unchanged. In this way program loops can be executed from the ROM even though Y is too short to address the ROM. The only way a program can jump from the ROM to the RAM is to load the program counter completely from a 16-bit memory location.

Some LISSYN instructions use Y as a numerical constant, in which case it is interpreted as a signed, 2's complement number and sign-extended to 16 bits.

IV. DETAILED STRUCTURE AND TIMING

A typical LISSYN instruction cycle can be followed with the aid of the central processor block diagram, Figure 3.

Each LISSYN instruction is implemented in two epochs. Epoch Ø is always 100 nsec (2 clock periods) long. At its start, the address of the next instruction is gated from the program counter, P, to the memory address input. When the instruction emerges from memory, its 6-bit OP code is decoded by the control memory into 32 control signals. The 10-bit numerical field of the instructions, if it refers to an address for reading or writing memory, is extended with zeros to 16 bits and sent to the ALU where it may be added to the contents of the index register, X, or pass through unchanged. At the end of epoch Ø, the resulting address is stored in a memory address register, MAR. At the end of epoch Ø, the OP code and the numerical field are latched into the OP and Y registers, respectively. Conditional jump instructions, which do not require computation of memory address, use the ALU during epoch Ø

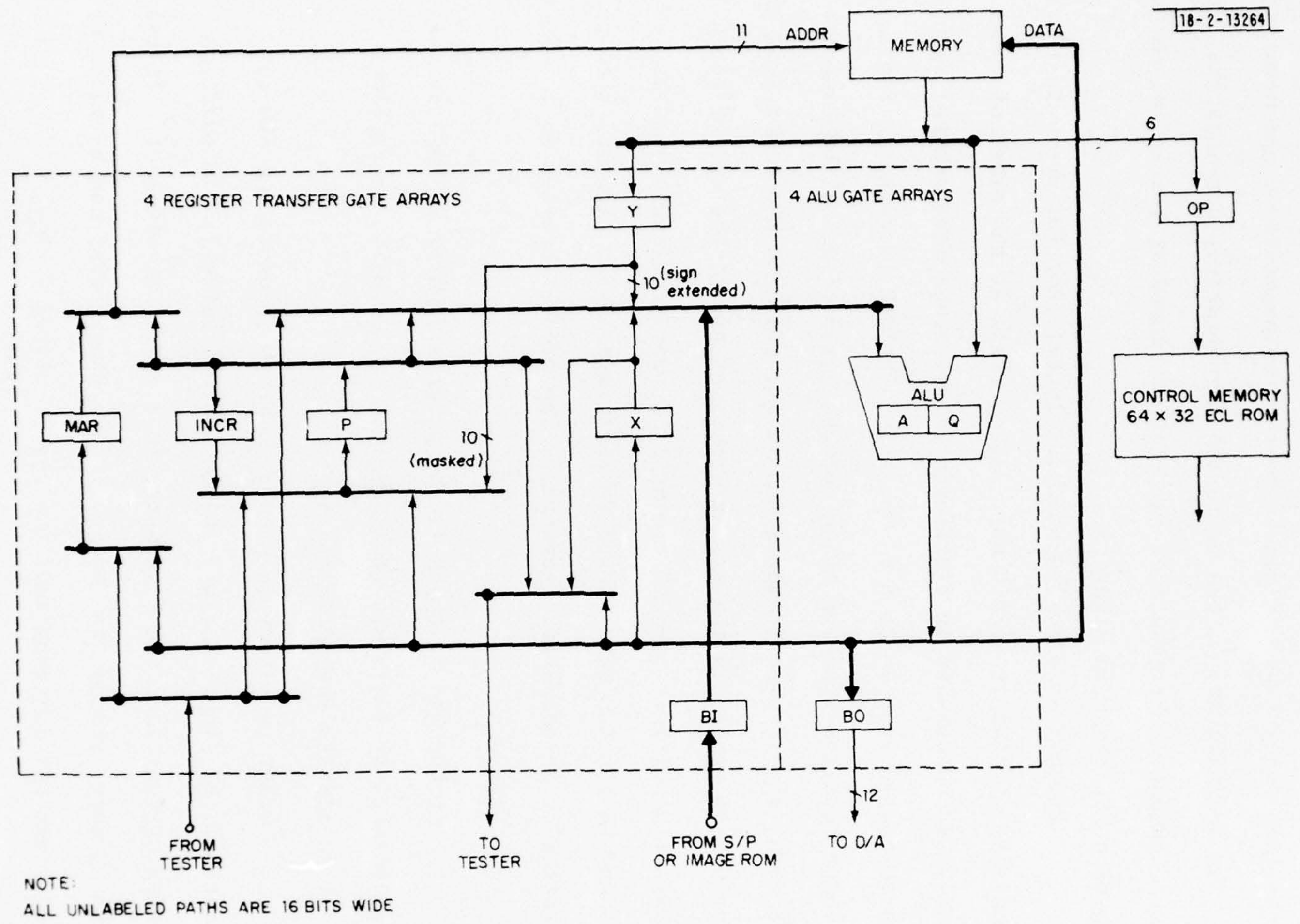


Fig. 3. Data paths in the *LISSYN* central processor.

to test the jump conditions. X can be tested for negative or non-negative content. The accumulator, A, can be tested for negative, non-negative, zero, or non-zero content. The result of any test is stored at the end of epoch Ø in a flip-flop not shown in Figure 3.

Epoch 1 is also 100 nsec (2 clocks) long for every instruction except multiply, when it is 850 nsec (17 clocks). At the start of epoch 1, the OP code is decoded again in a different mode, since some control signals will have to change from epoch Ø to epoch 1. The MAR is gated to the memory address input to allow fetching of an operand or storing of the contents of a register; then, any required computation (e.g., adding A and memory) is done, and the result is stored in the appropriate register at the end of epoch 1. In the case of a multiply, normal timing is interrupted and a sequence of 16 clocks is sent to the ALU gate arrays to perform a 16-bit signed 2's complement multiplication by an add/shift iteration. The product appears in the 32-bit combined A/Q register.

At the end of epoch 1, the program counter is updated. It can be incremented by 1 (INCR), replaced in bits 0-9 by Y, or replaced entirely by an address formed in the ALU.

Other registers in Figure 3 are Q (used in the ALU for multiplication), BI (input buffer) and BO (output buffer). In addition to its indexing function, the X register is a modest accumulator. It can be loaded from memory, stored in memory, and a memory word can be added or subtracted from its contents and the result stored in X.

V. THE TESTER

For the purposes of hardware debugging and software development, a separate box called the tester can be attached to the pair of buses shown in Figure 3 . The tester can do such tasks as: display the contents of P, X, A, or any main memory location; write any 16-bit word into any RAM location; display the instruction addressed by the P register; load the entire 1024-word RAM from a host computer; single-step through instructions; implement a hardware breakpoint. The LISSYN can run as an LPC synthesizer with the tester removed and, in that case, the LISSYN RAM is loaded from a 1024 x 16 TTL ROM (see Figure 1), via a bootstrap program in the 32-word LISSYN ECL ROM.

Most of the tester functions interface with the LISSYN in a novel fashion. The tester interrupts the LISSYN as its highest-priority peripheral device, causing it to branch to an address supplied by the tester. That address is the start of one of several short service routines located in the 32-word ECL ROM. These routines employ four special LISSYN instructions, which direct the LISSYN to transfer data to and accept data from the tester. This method of tester interfacing was adopted to save the multiplexers which otherwise would have been needed to allow the tester to force data and memory addresses onto LISSYN buses. It also allowed a single 16-bit cable to be used for both address and data when writing LISSYN memory from the tester, thereby saving scarce gate array pins.

VI. THE LPC SYNTHESIZER ALGORITHM

Figure 4 shows a block diagram of the LPC synthesizer algorithm. It accepts as input a serial bit stream produced in real time from speech by an LPC analyzer . The following description is for 4800 bits/sec, but programs for 3600 and 2400 bits/sec also exist. The information consists of

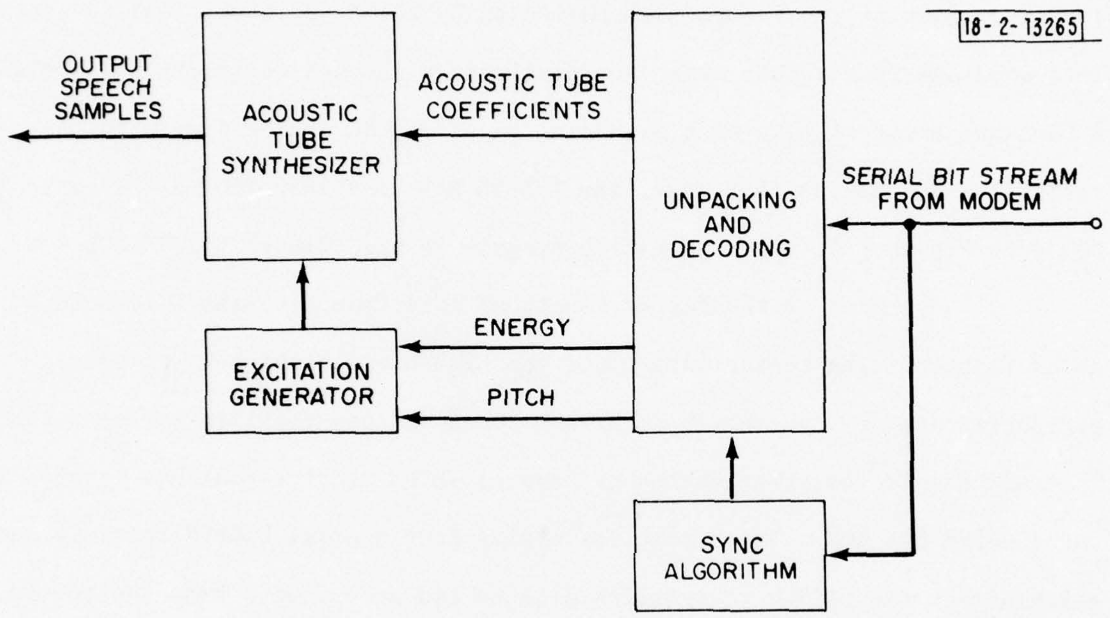


Fig. 4. The LPC synthesis algorithm.

14 binary code words describing the pitch (if speech is voiced), energy, and spectral shape of each frame (approximately 20 msec) of speech. If speech is unvoiced during a frame, the pitch word contains a bit pattern that allows synchronizing of the LISSYN with the bit stream, i.e., finding the start of a frame.

Once synchronization is established, the incoming code is unpacked and decoded. Two parameters, pitch and energy, are used to generate excitation for the acoustic tube, which models the vocal tract. The tube is specified by 12 coefficients, which are linearly interpolated approximately each 5 msec. Every 130 μsec, the acoustic tube generates another output speech sample.

Almost the entire computational load of the LPC synthesizer consists of cycling the acoustic tube. It requires 2 multiplies and 9 other instructions for each of the 12 tube coefficients for each sample of output speech. This takes

$$(2 \cdot 950 \text{ nsec} + 9 \cdot 200 \text{ nsec}) \cdot 12/130\ \mu\text{sec} = 34.2\% \text{ of real time.}$$

The entire 4800 bit/sec synthesis takes 43% of real time. The usage of memory is: program, 213 locations; constants (including decoding tables), 419 locations; dynamic storage, 85 locations.

VII. LISSYN ASSEMBLER AND SIMULATOR

Almost all of the logically complex functions of the LISSYN, particularly the I/O control, are buried inside gate arrays, where they are difficult to diagnose for failures and impossible to repair. For this reason, a detailed simulation of the system was needed to establish confidence that it could be made to work after assembly. The fact that each kind of gate array was simulated in MECL 10K before fabrication was useful in catching some design errors, but a full system simulation on a computer was still needed. The system simulation also allowed development of the LPC synthesizer programs while the LISSYN hardware was being built.

The LISSYN simulator runs on a Univac 1219 computer. It accepts as input the binary code generated by the LISSYN assembler, which also runs on the 1219. The simulator operates at quite a detailed level. Some functions are traced at the register level, others at the single-gate level. The machine state is updated at every clock period. Interrupts and data transfers between the LISSYN and its I/O devices are also simulated. The LPC synthesizer program was run on the simulator to the extent of entering sample frames of code and observing the waveform produced. One second of speech would take 24 hours to simulate.

As a result of the simulation, a redesign of the LISSYN I/O control was required. The corresponding gate array masks were changed before fabrication began, with a delay of only a day or two. The month of effort spent in developing the simulator program paid off handsomely. None of the five kinds of gate arrays had to be redesigned after their first fabrication. The LPC synthesis program ran the first time it was loaded into the LISSYN.

VIII. SUMMARY AND CONCLUSIONS

The Lincoln Integrated Speech Synthesizer (LISSYN), was designed to implement a specific LPC synthesis algorithm, as a demonstration of the ease and speed of realizing a complex logical system in custom-built ECL gate arrays. The whole system was designed and built in less than a year. The gate array production process is sufficiently flexible that a major change in one gate array design was made shortly before the mask-making stage with just a few days of added delay. Hardware simulation of each type of gate array and detailed software simulation of the overall system produced a reliable logical design despite the complexity of the gate arrays.

Virtually every part of the LISSYN that could have been made from gate array logic was included within the 11 gate arrays of 5 types. There were only 8 16-pin commercial packages of ECL logic gates. The rest of the packages were not suitable for integration onto gate arrays: 26 ECL memory packages, 1 ECL crystal-controlled clock, 21 packages using a +5V supply, 1 D/A converter.

Other parameters of the LISSYN are: single 1024 x 16 main memory for data and program, 200 nsec instruction cycle, 950 nsec add/shift multiply, binary serial input, analog output via 12-bit D/A converter and desampling filter, 0.35 cu. ft. volume, 60 watts DC power.

The goal of the LISSYN project was to produce a system with adequate computing power for LPC synthesis, with a minimum amount of hardware and engineering effort. The LISSYN runs the linear predictive speech synthesis in 43% of real time. However, this speed is much less than can be obtained from gate array logic, which has about 1.5 nsec/gate average delay. The addition of a faster multiplier rather than the add/shift system used for LISSYN would permit the use of pipelining techniques that could make the LISSYN at least twice as fast as it is.

The development of a fast 16-bit multiplier, small enough to reside in a LISSYN type machine without significantly increasing its overall size or power consumption, appears to be possible using dielectric isolation techniques. A reasonable extension of the LISSYN project would therefore be the fabrication of such a multiplier and its subsequent inclusion in either a full-duplex gate array vocoder or a multiple-synthesizer version of the LISSYN.

APPENDIX A
DESCRIPTION OF LISSYN GATE ARRAYS

1) ALU 4-Bit Slice

Figure A1 shows a block diagram of the ALU gate array. The heavy lines represent 4-bit data paths. There are three 4-bit registers of master-slave flip-flops. The B register is simply an output buffer for the adder output, which also appears directly on the S bus. A and Q are general registers that can provide input to and store output from the adder. The shifted inputs to A and Q are chosen to facilitate linking A and Q for add/shift multiplication. There is a zero detector on the output of the A register. External data can enter on two 4-bit buses, L and R. The usual 2's complement arithmetic operations and logical operations are available.

The ALU gate array has greater capability than was needed for the LISSYN. For example, the LISSYN makes no use of the ability to add or subtract Q from memory, detect overflow (OVF), and provide group propagate (GP) and generate (GG) signals for carry look-ahead.

2) Register-Transfer 4-Bit Slice

The block diagram of 4 linked register-transfer gate arrays appears in Figure 3 and their operation is described in section IV . One feature not appearing in the linked drawing is carried in and out for the incrementer of the program counter. Another is two special control lines that allow the upper 2 bits to be differentiated from the lower two bits in order to implement the 10-bit masking and sign extension of the Y field, whose boundary does not coincide with the boundary of a 4-bit slice.

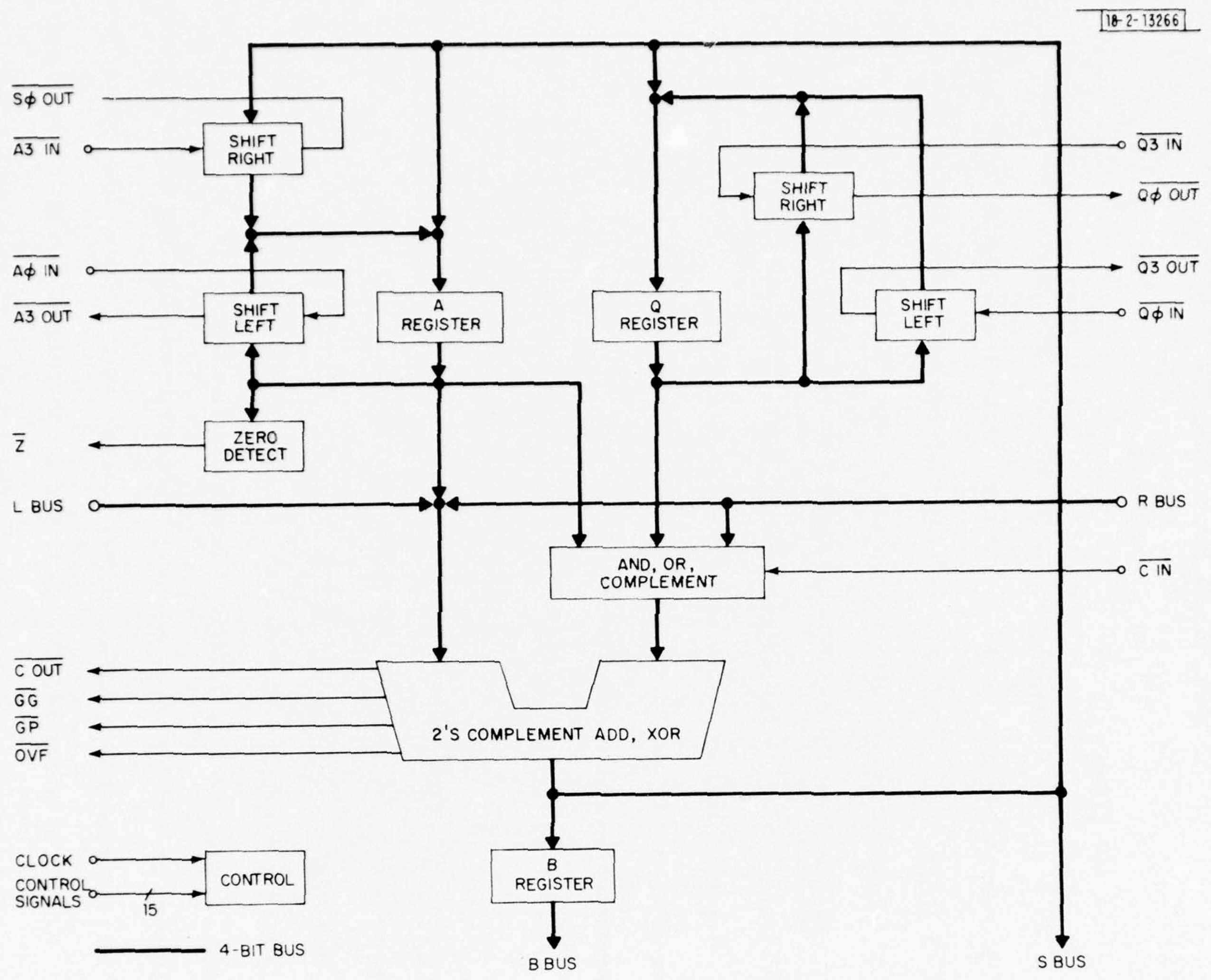


Fig. A1. The ALU gate array.

3) Control A

The Control A gate array performs four nearly separate control functions. See Figure A2 .

Timing Generator - This circuit receives as input the 20 MHz oscillator and three of the timing phases produced by the phase generator gate array. It produces timing signals for the rest of the LISSYN. ACLK clocks the ALU gate array. EP1 and EPØ define the two instruction epochs and clock almost everything else in the LISSYN. (The phase generator and a few flip-flops are clocked directly by the oscillator.) In addition to clocks, the timing generator produces signals to start or stop the phase generator on the multiply (MUL) and halt (HLT) instructions and multiply phases that are used elsewhere in Control A to distinguish among the first, last, and remaining multiplier bits.

End Logic - This circuit examines the appropriate arithmetic status bits and computes the proper value to shift into the high order bit of the A register. Most of its complexity stems from the special treatment of the sign bit in the 2's complement add/shift multiply.

OP Code Latch and Conditioner - This circuit stores the OP code during epoch 1 of the instruction cycle and generates the conditional OP code for addressing the control ROM. See Appendix B, Instruction Decoding.

Control Modification A - This circuit modifies 5 outputs of the control ROM, mostly to aid in multiplies and index additions. See Appendix B, Instruction Decoding.

4) Control B

The Control B gate array performs five nearly separate functions. See Figure A3 .

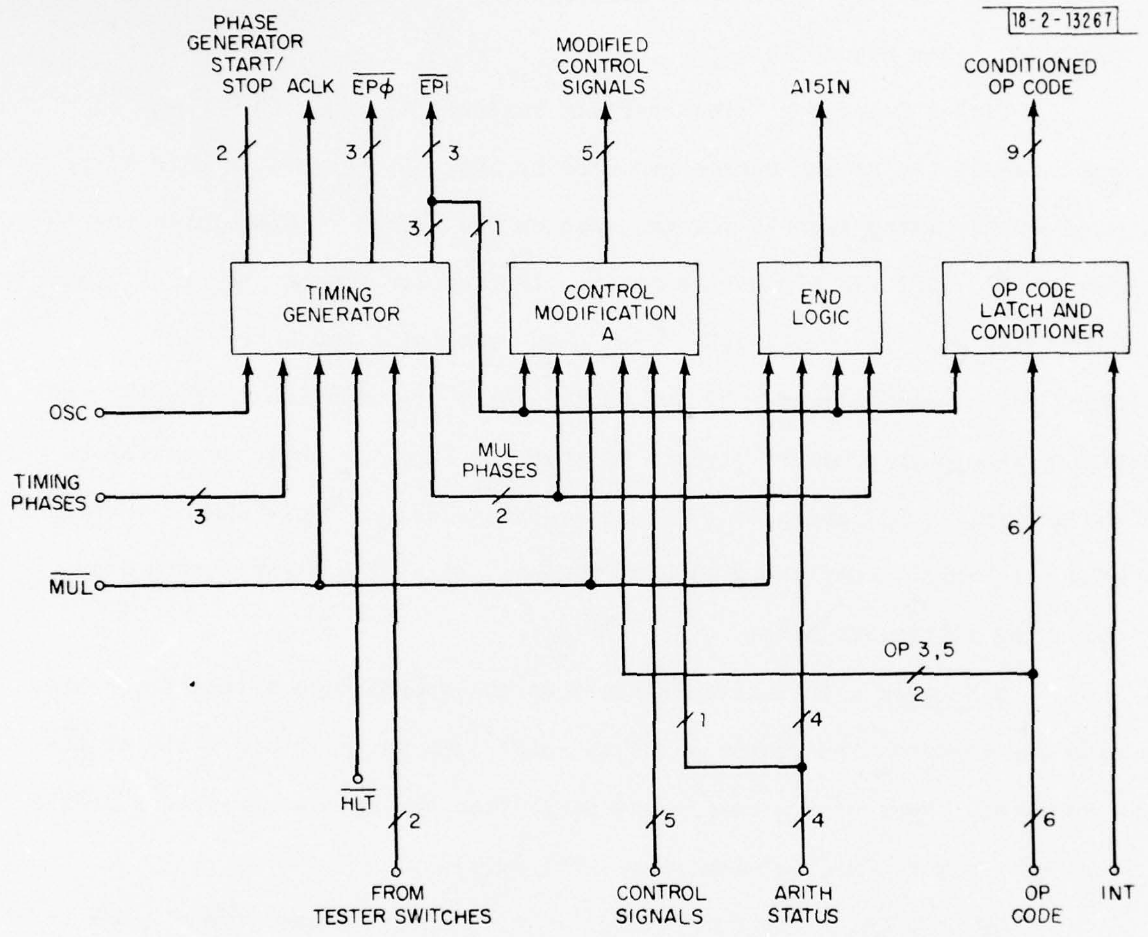


Fig. A2. The control A gate array.

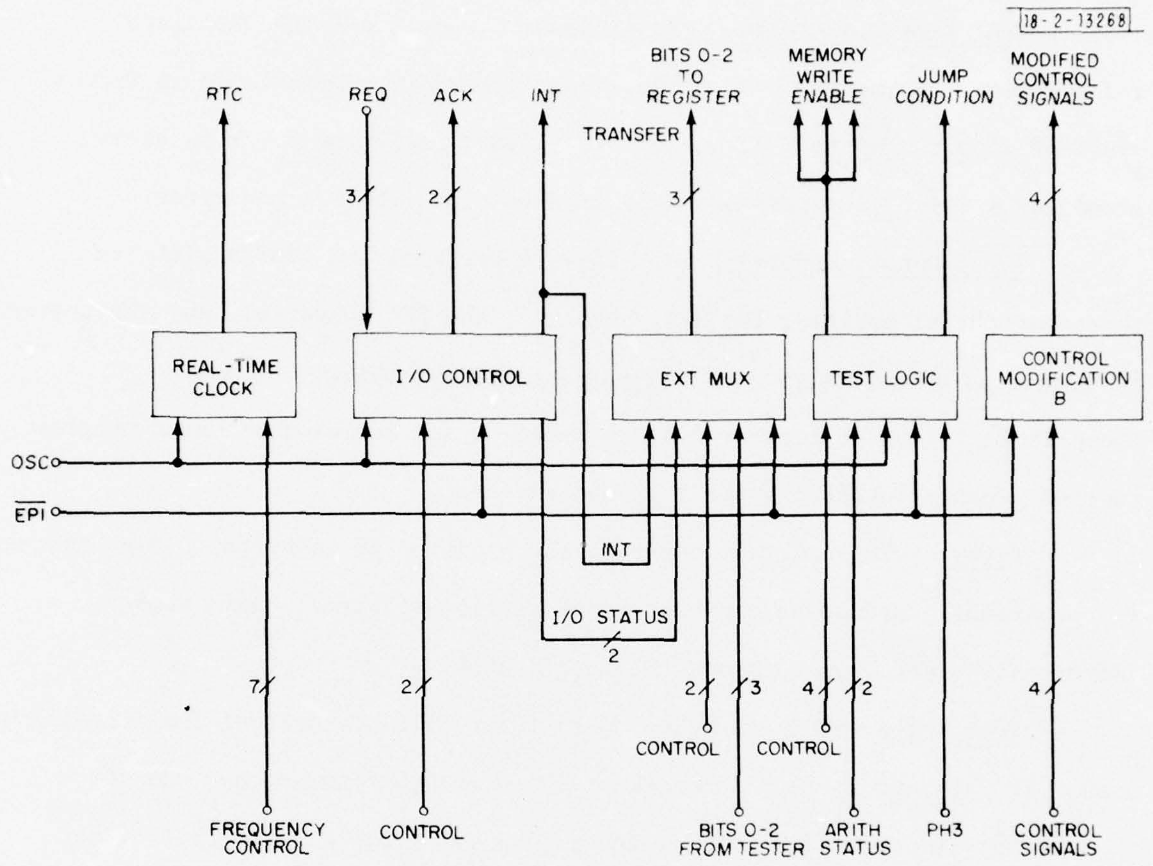


Fig. A3. The control B gate array.

Real-Time Clock - This circuit runs directly off the oscillator and produces a symmetric square wave whose period is controllable in increments of 1.6 μsec up to 203.2 μsec. The LISSYN real-time clock is hard-wired for a 129.6 μsec period and is used to clock the D/A converter.

Input/Output Control - *This logic interfaces the* LISSYN with its three peripheral devices, the D/A converter, the S/P converter, and the tester. It generates an interrupt (INT) signal and two I/O status signals. There is an interrupt lockout flag that can be set and cleared under program control. Tester interrupts can't be locked out.

Extmux - This circuit controls the handling of entry and return addresses for programmed subroutines and interrupt service routines. Main memory addresses 0-4 are used to store these addresses.

Test Logic - This circuit stores the result of any test for a conditional jump. If the jump is to a subroutine, the writing of the return address is also conditional. The outputs are a jump condition bit to control the updating of the program counter and a fanned-out memory write enable capable of driving all the RAM chips.

Control Modification B - This circuit modifies 4 outputs of the control ROM, mostly to aid in index addition. See Appendix B, Instruction Decoding.

PHASE GENERATOR

See Figure A4. This circuit converts the 20 MHz oscillator signal into four timing phase signals, PHØ, PH1, PH2, PH3 which are then used for gating other functions on the Control A and Control B gate arrays. The phase generator can be started, stopped, single-stepped, or

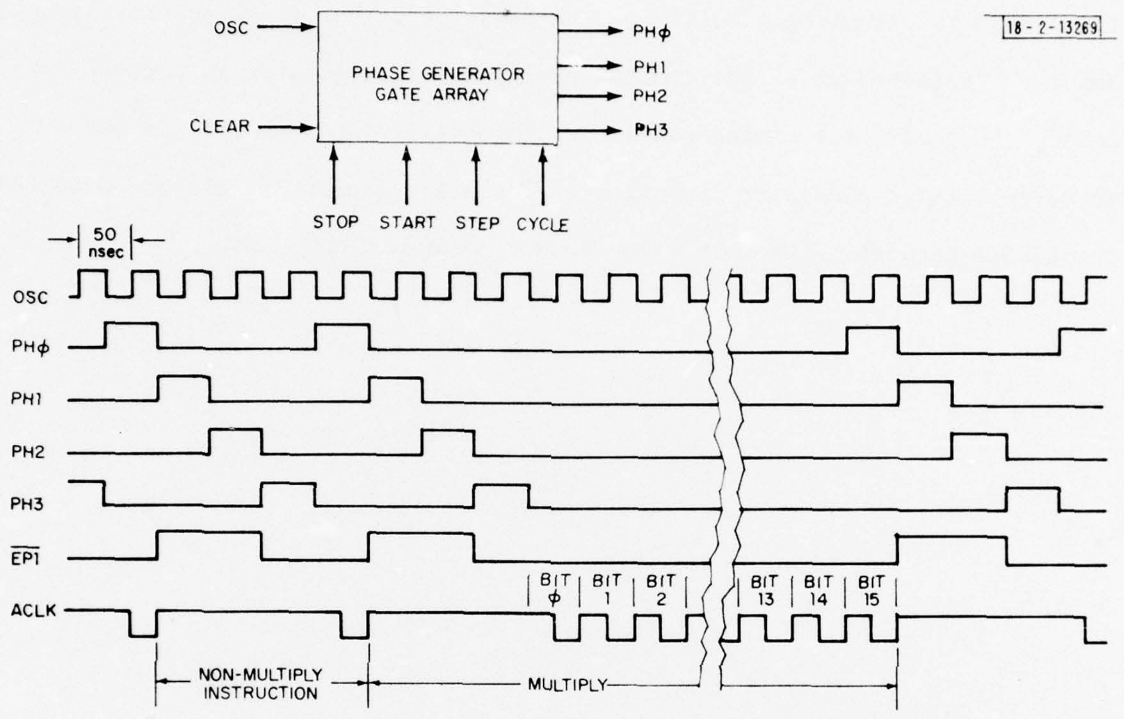


Fig. A4. The phase generator gate array.

cycled to a known starting state under the control of signal lines. When the LISSYN is executing a multiply, the phase generator is temporarily stopped and then restarted under the control of the timing generator in Control A. During the pause, the timing generator produces 16 ACLK signals for the ALU to carry out the add/shift iterations. The phase-generator design predated the LISSYN project. The chip area is less than half utilized.

APPENDIX B

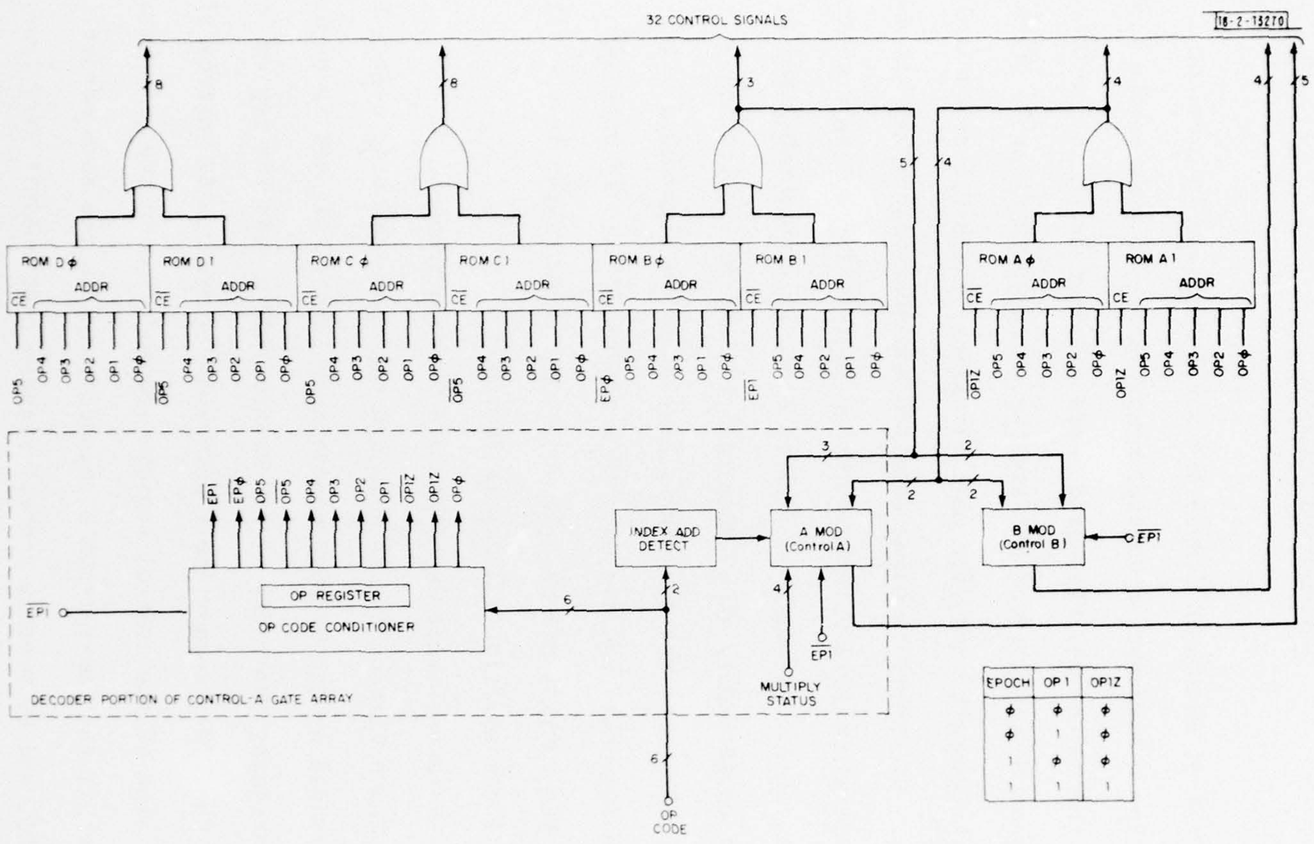
INSTRUCTION DECODING

A tradeoff of increased complexity for a decrease in critical decoding delay and a decrease in control memory size was made in the LISSYN. The pattern of the 32 control signals depends not just on the 6-bit OP code of the instruction being executed but also on which epoch (Ø or 1) of that instruction is in progress and, for multiplication, which bit of the multiplier (low-order, sign, or other) is being examined and what value that bit has. If all these parameters were used to form the address for a single control memory, it would need 1024 words of 32 bits each, an unacceptable amount. In addition, it would unduly slow index addition and multiplication. Therefore the LISSYN uses the decoding scheme shown in Figure B1 . It is less regular in form but requires only 64 words of 32 bits (8 packages of 32 x 8) for the control memory. In addition, it is faster in the critical delay paths of index addition and multiplication.

Figure B1 reveals that decoding is a 3-stage process.

1. The 6 bits of the OP code and the epoch-defining signal EP1 are expanded by the OP code conditioner into 11 signals to be used for addressing the eight 32 x 8 control ROM's near the top of Figure B1 .
2. The ROM's decode their addresses to produce 32 control signals.
3. Nine of the control signals are modified in the gate arrays.

Notice that not all the control ROM's are addressed in the same manner. ROM's CØ, C1, DØ, and D1 produce control signals that needn't vary from epoch Ø to epoch 1. These can use the 6 bits of the OP code for addressing



EPOCH	OP 1	OP1Z
ϕ	ϕ	ϕ
ϕ	1	ϕ
1	ϕ	ϕ
1	1	1

Fig. B1. LISSYN instruction decoding scheme.

in the usual manner. ROM's BØ and B1 produce control signals which do vary from epoch Ø to epoch 1. However. careful assignment of OP codes permits the control signal pattern to be insensitive to OP code bit 2 during epoch Ø and to bit 3 during epoch 1. In each epoch, a pair of instructions effectively shares each word in a ROM, but the pairing is different in the two epochs. ROM's AØ and A1 produce control signals that vary between epoch Ø and epoch 1 when OP code bit 1 is a 1 but do not vary between epochs when OP code bit 1 is a 0. In epoch Ø a pair of instructions share each word of ROM AØ, but in epoch 1, each instruction has its own word in ROM AØ or ROM A1.

The modifications needed for index addition are such that the critical outputs of the control modification A block are independent of the inputs from the ROM's, so the delay through the ROM's is avoided. During multiplication, the ROM outputs do not change with the position of the bit being processed, so again the delay through control modification A is small.

APPENDIX C

THE LISSYN INSTRUCTION SET

Each LISSYN instruction has a 6-bit OP code and a 10-bit numerical field, Y. The following abbreviations have been used in the instruction table:

Y	The ten-bit Y field, interpreted as a positive integer, extended with zeros to 16 bits.
Y,sx	The ten-bit Y field, interpreted as a signed, 2's complement integer, sign extended to 16 bits.
Y,msk	The ten-bit Y field replaces bits 0-9 of the register being altered. Bits 10-15 of that register are unchanged.
ILO	The interrupt lockout flag.
TØ	The 16 bits on the tester bus to the LISSYN at the end of instruction epoch Ø.
T1	The 16 bits on the tester bus to the LISSYN at the end of instruction epoch 1.
M(n)	The n-th location of main memory
+	2's complement addition
—	2's complement subtraction or negation
$\cdot$	Signed 2's complement multiplication
$\vee$	logical OR
$\wedge$	logical AND
$\oplus$	logical exclusive OR
$\overline{R}$	logical complement of register R

PROGRAMMED JUMPS

OCTAL OP CODE	MNEMONIC	CONDITION AT START OF INSTRUCTION	REGISTER OR FLAG ALTERED	NEW VALUE	OTHER INFORMATION
0ϕ	JPZAS	$A \geq 0$	P M(0)	Y, msk P	conditional jump to subroutine
		$A < 0$	P	P + 1	
ϕ1	JNAS	$A < 0$	P M(0)	Y, msk P	conditional jump to subroutine
		$A \geq 0$	P	P + 1	
ϕ4	JZAS	$A = 0$	P M(0)	Y, msk P	conditional jump to subroutine
		$A \neq 0$	P	P + 1	
ϕ5	JUZAS	$A \neq 0$	P M(0)	Y, msk P	conditional jump to subroutine
		$A = 0$	P	P + 1	
ϕ6	IOIJP		P	M(1)+Y,sx	return from I/O subroutine
ϕ7	JPS		P	Y, msk	
1ϕ	JPZA	$A \geq 0$	P	Y, msk	
		$A < 0$	P	P + 1	
11	JNA	$A < 0$	P	Y, msk	
		$A \geq 0$	P	P + 1	
12	JPZX	$X \geq 0$	P X	Y, msk X - 1	
		$X < 0$	P X	P + 1 X - 1	
13	JNX	$X < 0$	P X	Y, msk X + 1	
		$X \geq 0$	P X	P + 1 X + 1	

PROGRAMMED JUMPS (Cont'd)

OCTAL OP CODE	MNEMONIC	CONDITION AT START OF INSTRUCTION	REGISTER OR FLAG ALTERED	NEW VALUE	OTHER INFORMATION
14	JZA	A = 0	P	Y, msk	
		A ≠ 0	P	P + 1	
15	JUZA	A ≠ 0	P	Y, msk	
		A = 0	P	P + 1	
16	IJP		P	M(0)+Y,sx	return from subroutine
17	JP		P	Y, msk	
25	JPCON		P	T1	Used to start LISSYN at address in tester switches

MISCELLANEOUS CONTROL INSTRUCTIONS

∅2	INT	D/A Interrupt	M(1) P	P 2	Not recommended as a programmed instruction. Result when no interrupt is present is not uniquely known.
		S/P Interrupt	M(1) P	P 3	
		Tester Interrupt	M(4) P	P T1	
2∅	HLT		none		LISSYN stops execution. If start switch is pushed, the next instruction will be taken from P+1.
21	SIL		ILO P	1(set) P + 1	
34	RIL		ILO P	∅(cleared) P + 1	

MEMORY READ/WRITE

22	STCON		M(T∅) P	T1 P + 1	Used to write memory from tester

MEMORY READ/WRITE (Cont'd)

OCTAL OP CODE	MNEMONIC	CONDITION AT START OF INSTRUCTION	REGISTER OR FLAG ALTERED	NEW VALUE	OTHER INFORMATION
24	AXCON		P	P + 1	A or X register is displayed at tester, according to switch on tester.
32	YIX		X P	Y, sx P + 1	
35	LDCON		P	P + 1	M(TØ) is displayed at tester
47	LDQX		Q P	M(Y+X) P + 1	
57	LDQ		Q P	M(Y) P + 1	
60	STAX		M(Y+X) P	A P + 1	
70	STA		M(Y) P	A P + 1	
61	STXX		M(Y+X) P	X P + 1	
71	STX		M(Y) P	X P + 1	
62	STBX		M(Y+X) P	BI P + 1	
72	STB		M(Y) P	BI P + 1	
63	STPX		M(Y+X) P	P P + 1	
73	STP		M(Y) P	P P + 1	
64	LDAX		A P	M(Y+X) P + 1	

MEMORY READ/WRITE (Cont'd)

OCTAL OP CODE	MNEMONIC	CONDITION AT START OF INSTRUCTION	REGISTER OR FLAG ALTERED	NEW VALUE	OTHER INFORMATION
74	LDA		A P	M(Y) P + 1	
65	LDXX		X P	M(Y+X) P + 1	
75	LDX		X P	M(Y) P + 1	
66	LDBX		BO P	M(Y+X) P + 1	
76	LDB		BO P	M(Y) P + 1	
67	LDPX		P	M(Y+X)	
77	LDP		P	M(Y)	

ARITHMETIC/LOGICAL INSTRUCTIONS

Ø3	CLA		A P	Ø P + 1	
23	SUBX		X P	X-M(Y) P + 1	
33	ADDX		X P	X+M(Y) P + 1	
4Ø	ADDAX		A P	A+M(Y+X) P + 1	
5Ø	ADDA		A P	A+M(Y) P + 1	
41	AXORX		A P	A ⊕ M(Y+X) P + 1	
51	AXOR		A P	A ⊕ M(Y) P + 1	
42	SUBAX		A P	A-M(Y+X) P + 1	

ARITHMETIC/LOGICAL FUNCTIONS (Cont'd)

OCTAL OP CODE	MNEMONIC	CONDITION AT START OF INSTRUCTION	REGISTER OR FLAG ALTERED	NEW VALUE	OTHER INFORMATION
52	SUBA		A P	A-M(Y) P + 1	
43	MMAX		A P	M(Y+X)-A P + 1	
53	MMA		A P	M(Y)-A P + 1	
44	AANDX		A P	$A \wedge M(Y+X)$ P + 1	
54	AAND		A P	$A \wedge M(Y)$ P + 1	
45	MULX		A Q P	Q·M(Y+X), bits 16-31 Q·M(Y+X), bits Ø-15 P + 1	Takes 950 nsec
55	MUL		A Q P	Q·M(Y), bits 16-31 Q·M(Y), bits Ø-15 P + 1	Takes 950 nsec
46	AORX		A P	$A \vee M(Y+X)$ P + 1	
56	AOR		A P	$A \vee M(Y)$ P + 1	
26	HVAQ		A, Q P	(A,Q)/2 P + 1	Linked right shift with sign extension
27	DBAQ		A, Q P	2(A, Q) P + 1	Linked left shift
30	CMPA		A P	$\overline{A}$ P + 1	

ARITHMETIC/LOGICAL FUNCTIONS (Cont'd)

OCTAL OP CODE	MNEMONIC	CONDITION AT START OF INSTRUCTION	REGISTER OR FLAG ALTERED	NEW VALUE	OTHER INFORMATION
31	STQA		A P	Q P + 1	Used to retrieve low-order product. Q cannot be stored in memory.
36	CSA		A P	-A P + 1	
37	DBA		A P	2A P + 1	

ACKNOWLEDGMENTS

The author gratefully acknowledges the guidance in logical design and packaging provided by K. Konkle, the development of simulators and test programs contributed by A. Soares, the mechanical design done by E. Aho, and the construction and debugging assistance of J. Fisher.

REFERENCES

1. J.D. Markel and A.H. Gray, Jr., Linear Prediction of Speech (Springer-Verlag, New York, 1976).

2. E.M. Hofstetter *et al.*, "Vocoder Implementation on the Lincoln Digital Voice Terminal," EASCON '75, Washington, D.C., 29 September-1 October 1975.

3. J.I. Raffel *et al.*, "A Flexible, Subnanosecond, ECL Gate Array," Government Microcircuit Applications Conference, Orlando, Florida, 9-11 November 1976.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 **REPORT DOCUMENTATION PAGE**	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 18 ESD-TR-77-58 — 2. GOVT ACCESSION NO. 14	3. ~~RECIPIENT'S CATALOG NUMBER~~ TN-1977-11
4. TITLE (and Subtitle) 6 An ECL Gate Array Realization of a Computer for Real Time Speech Synthesis.	5. ~~TYPE OF REPORT & PERIOD~~ COVERED 9 Technical Note,
	6. PERFORMING ORG. REPORT NUMBER Technical Note 1977-11
7. AUTHOR(s) 10 Robert Berger	8. CONTRACT OR GRANT NUMBER(s) 15 F19628-76-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element No. 65705F Project No. 649L 16
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Systems Command, USAF Andrews AFB Washington, DC 20331	12. REPORT DATE 11 7 March 1977
	13. NUMBER OF PAGES ~~46~~ 12 44p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB Bedford, MA 01731	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ECL gate array	speech processing	Integrated Speech Synthesizer
real time speech	linear predictive vocoder	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This note describes the Lincoln Integrated Speech Synthesizer (LISSYN), a general-purpose computer intended for speech processing, whose central processor is made from ECL gate arrays (large scale integrated circuits custom built at Lincoln Laboratory).

The goal was to use gate arrays to implement in real time the synthesis portion of a linear predictive vocoder operating at 4800 bits/sec. The design process stressed minimizing the number of different kinds of gate arrays and the number of non-gate-array circuit packages. The result is a general purpose computer structure featuring: single 1024 × 16 memory for data and program, 200 nsec instruction cycle, 950 nsec add/shift multiply, binary serial input, analog output via a 12-bit D/A converter and desampling filter, 0.35 cu. ft. volume, 60 watts DC power, 11 gate arrays of 5 types, 30 memory IC's, 27 other circuit packages. The LISSYN runs the linear predictive speech synthesis in 43% of real time.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

207650 [illegible]